



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/204,971	12/03/1998	DAVID LARS EHNEBUSKE	AT9-98-267	9311

7590 07/31/2002
DUKE W YEE
CARSTENS YEE & CAHOON
P O BOX 802334
DALLAS, TX 75380

EXAMINER

INGBERG, TODD D

ART UNIT PAPER NUMBER

2124

DATE MAILED: 07/31/2002

Please find below and/or attached an Office communication concerning this application or proceeding.



UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

MAILED
JUL 30 2002
Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Paper No. 17

Application Number: 09/204,971
Filing Date: December 03, 1998
Appellant(s): EHNEBUSKE ET AL.

Stephan J. Walder, Jr.
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed May 2, 2002.

Art Unit: 2124

(1) *Real Party in Interest*

A statement identifying the real party in interest is contained in the brief.

(2) *Related Appeals and Interferences*

The brief does not contain a statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief. Therefore, it is presumed that there are none. The Board, however, may exercise its discretion to require an explicit statement as to the existence of any related appeals and interferences.

(3) *Status of Claims*

The statement of the status of the claims contained in the brief is correct.

(4) *Status of Amendments After Final*

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) *Summary of Invention*

The summary of invention contained in the brief is correct.

(6) *Issues*

The appellant's statement of the issues in the brief is correct.

(7) *Grouping of Claims*

Group I - Claims 1, 7, 10, 12, 19, 24 and 25 stand or fall together .

Group II – Claims 8, 22 and 28 stand or fall together .

Group III – Claims 2, 3, 9, 20, 21, 23, 26, 27 and 29 stand or fall together .

Group IV – Claims 4 and 16 stand or fall together .

Art Unit: 2124

Group V – Claims 5 and 17 stand or fall together .

Group VI – Claims 6 and 18 stand or fall together .

Group VII - Claim 11 stands or falls.

Group VIII – Claim 13 stands or falls.

(8) *Claims Appealed*

The copy of the appealed claims contained in the Appendix to the brief is correct.

(9) *Prior Art of Record*

The copy of the appealed claims contained in the Appendix to the brief is correct.

(10) *Grounds of Rejection*

Claims 1 – 29 are rejected under 35 U.S.C. 103(a). This rejection is set forth in prior Office Action, Paper No. 12 is the Final Rejection with complete response.

(11) *Response to Argument*

Appellant's Statement's on Not Providing an Information Disclosure

“The issues regarding the filing of an Information Disclosure Statement and Definitions of Terms raised in the First Office Action have been withdrawn by the Examiner. However, Appellants, in view of the Examiner's statements in the first twenty pages of the Final Office Action with regard to these issues, wish to again reiterate that Appellants are under no obligation to research alleged prior art raised by the Examiner nor supply information regarding prior art raised by the Examiner of which Appellants are not aware of any materiality with regard to the patentability of the present invention, as defined by the pending claims. If the Examiner is aware of alleged prior art that the Examiner believes is material to patentability, the Examiner must cite the art on a PTO Form 1449 and cannot shift the burden to Appellants simply by raising the issue in an Office Action.”

Examiner's Response

It is true that to date, **all** the prior art of record has been cited by the Examiner and that the Examiner as he found numerous references to the Assignee's (IBM) work. Furthermore, the Examiner has inquired and provided all material in his possession. The Examiner must assume the inventor and Assignee have acted in good faith and condor before the office.

Additionally, the Examiner withdrew the rejection under 35 U.S.C. 112 when he accepted the terms and conditions as a result of prosecution (FAOM and Response to Office action). The interpretation of terms was cooperation between the Examiner and the now Appellant.

Art Unit: 2124

Appellant's response added clarity to some of the terms and reinforced the Examiner's interpretations of other terms and was instrumental in overcoming the rejection under 35 USC 112. The record is clear as to the interpretation given to the terms in the prosecution. Somehow the Appellant is trying to state terms were withdrawn. This is not factual. The Examiner found the response supported the terms the Examiner had raised or added clarity and in the Appellant's own words. The record is very clear on the definitions given to terms in the prosecution. If the Appellant takes issue then clear, concise and timely response was needed to traverse an interpretation. To date no such traversal is of record and the Appeal Brief also fails to properly argue the interpretation of the terms. In full view of the well built record the Appellant makes every attempt to distance themselves from the terms in prosecution history. However, the Appellant failed in all instances to point to the Specification and provide clear and concise definitions with proper technical support from the Specification.

* The Board of Patent Appeals and Interference (BPAI) should refer to the Final Action for the record of the interpretation of terms.

Appellant's Additional Remarks about Terms

"Furthermore, with regard to the definitions of terms asserted by the Examiner, again Appellants wish to reiterate that the terms in the pending claims are not limited by the Examiner's asserted definitions. Appellants have offered other example definitions of these terms as taken from the present specification. These definitions are not meant to be limiting with regard to interpretation of the claims but are only offered as examples. Similarly, any alleged agreement by Appellants with the Examiner's asserted definitions is only an agreement that the Examiner's definitions are adequate examples of how these terms may be interpreted in view of the specification. Such agreements and definitions are not "findings of fact" and are not intended to simply any limitations with regard to interpretation of the claims."

Examiner's Response

Ultimately in patent law words mean things in Patent Law to the extent that the Applicant is not providing a definition and the Examiner has provided meaning throughout the prosecution history. Those are ultimately a finding of fact.

The record is very clear of the terms and the definitions given in the prosecution of the case. Many terms were given the interpretation from the Appellant's own words. Furthermore, the Appellant's response (paper # 11) was given praise by the Examiner and successfully overcame a rejection under 35 U.S.C. 112. The cooperation of the Appellant and Examiner has provided a clear record of the meaning of the terms in the prosecution of this Patent application. If the Appellant wishes to distance themselves from the definition of the prosecution the Appellant should have provided sound technical support from the Specification accompanied by technical arguments. Both are lacking throughout the following.

II. Appellant's Argument on the Grounds of Rejection

Appellant's Generalization of the Rejection

Art Unit: 2124

35 U.S.C. § 103 (a) Obviousness

"The Final Office Action rejects claims 1-29 under 35 U.S.C. § 103(a) over Martin, Principles of Object-Oriented Analysis and Design, 1993 in view of alleged common knowledge of programming as taught by Code Complete and definitions as provided by the IBM Computer Dictionary. The rejection is primarily based on Martin with Code Complete and IBM Computer Dictionary allegedly providing the teachings of using "common sense" in programming and some well-known terminology. This rejection is respectfully traversed."

Examiner's Response

The Martin reference which the Appellant elects to paint as a general teaching, teaches in addition to foundation knowledge on Object-Oriented technology, specifically teaches the underlying theory of developing Computer Aided Software Engineering (CASE) tools. The Martin book can not be considered a general teaching on Object Oriented technology just because it teaches the principles of the technology. The book's main focus is on the specific teaching of CASE tools. A general teaching on Object Oriented technology does not cover CASE tools or OO-CASE tools. The Appellant's painting of the Martin reference as a general teaching shows a lack of understanding of what is a general teaching on Object-Oriented technology. The focus of the rejection is Martin's teaching on OO-CASE tools. Martin teaches several mechanics to deploy such tools including flowchart based workflow systems that generate executable code. The Martin reference covers most of the limitations of the claimed invention.

Group I - Patentability for claims 1, 7, 10, 12, 19, 24 and 25 hinge on claim 1**The actual rejection is as follows.****Claim 1**

Martin teaches a method for performing general integrity checks using rules in an application running on a data processing system (Martin, page 133, 138 - 142, structure of Rules) comprising: identifying a point in a unit of work where application state integrity is to be verified (Martin, page 143, object state rules), wherein the unit of work includes a plurality of participants (Martin, page 143, last paragraph, linked to appropriate items in OO diagram); obtaining rules associated with each participant in the unit of work (Martin, page 144, Box 10.3); responsive to obtaining the rules (Martin, page 136, Rules Linked to Diagrams), running the rule obtained for each of the participants to verify the integrity of an application state (Martin, page 143, object state rules), according to the plurality of participants; general integrity

Art Unit: 2124

checks running on a data (Martin, page 133, 138 - 142, structure of Rules). Martin does not explicitly teach the programming constructs of responsive to determining that the unit of work is to be completed. Although, Martin clearly supports Rules and the testing for conditions the reference does not explicitly state RULES can be used to determine if a write operation should be performed or not. The following overviews the teaching of Martin.

Martin Reference teaches RULES

Testing the Integrity of an Object (page 143)

When condition (Chapter 10 a format for Rules logic structure testing the state of the Object)

Perform operation

When condition

Perform operation

(Note: the construct closely resembles the structure of a CASE/switch statement in many languages)

What Martin does give is the endless possibilities of the conditions and the endless possibilities of the operations, such as if a state condition is wrong don't write (negative results) if the state operation is correct (positive results) perform a write operation. Examiner, holds determinations when to perform a write operation and when not to perform a write are issues of normal use and considered part of being a artisan of ordinary skill in the art. The term for employing common sense in programming is called Defensive Programming. It is the reference Code Complete by Steve Mc Connell that teaches defensive programming like on page 97 of Code Complete A Garbage In Does Not Mean Garbage Out. In other words test before you use data. Therefore, it would have been obvious to one of ordinary skill at the time of invention to use the teaching of Martin's RULES to perform defensive programming, because " ... it's the recognition that programs will have problems and modifications, and that a smart programmer will develop code accordingly. (Code Complete, page 94, 5.6 Defensive Programming, Key Point)

Appellant's Argument

"Claim 1, which is representative of claims 19 and 25 with regard to similarly recited subject matter, reads as follows:

1. A method for performing general integrity checks using rules in an

Art Unit: 2124

application running on a data processing system comprising:

- identifying a point in a unit of work where application state integrity is to be verified, wherein the unit of work includes a plurality of participants;
- responsive to determining that the unit of work is to be completed,
- obtaining rules associated with each participant in the unit of work; and
- responsive to obtaining the rules, running the rules obtained for each of the participants to verify the integrity of an application state,' according to the plurality of participants."

It should first be noted that the Martin reference is a reference comprising over 400 pages of text. Therefore, Appellants are only addressing those sections explicitly cited by the Final Office Action. Although the Examiner asserts, in the Final Office Action, that the reference must be taken as a whole, it is the Examiner's burden to cite with specificity, the sections of the reference that the Examiner believes support a finding of obviousness. It is not Appellants' duty to delve through 400 pages of text to recreate the Examiner's reasoning. Such a requirement would shift the burden of establishing a prima facie case of obviousness from the Examiner and place it on Appellants. Thus, Appellants are addressing the teachings of Martin as explicitly cited by the Examiner. Moreover, even taking Martin as a whole, the reference does not teach any of the features of the present invention, as detailed hereafter. Martin is a general textbook that includes a section on the use of rules in object-oriented programming. While Martin generally teaches the use of rules, and even integrity rules, Martin does not teach or even suggest the specific use of rules set forth in claims 1, 19 and 25. A general teaching does not render every specific use of the general teaching obvious. The following are Appellants' responses to the Final Office Action's allegations regarding the specific features recited in claims 1, 19 and 25. Martin does not teach or suggest any of these features."

Examiner's Response

The rejection is very explicit on the topic of OO-CASE tools and meets the level of the disclosure. First, the anatomy of the rule in the rejection is from Chapter 10 page 133. The testing of the integrity of an object is on page 143 as given in the rejection. On the cited pages the rejection clearly mentions the code is generated from the diagrams/models (page 133 and 135) and that the rules are linked to the diagrams (page 136).

Page 137 of the rejection shows the category of Rules including the Integrity Rule. Also on page 147 the integrity rule is shown in use with a state transition diagram.

The Martin reference clearly teaches rules and integrity checking as cited by the rejection. The Appellant has not made a distinction over the prior art but has made allegations of patentability without technical merit provided. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

Basic knowledge of object oriented technology as one of ordinary skill in the art would have known at the time of invention is all that is need to follow the reasoning of the rejection. The burden was shifted from the Examiner to the Appellant with the FAOM. The Appellant failed to provide technical reasoning with support from the Specification on why the claimed invention differs from the prior art of record.

Art Unit: 2124

The argument that Martin does not teach specific use of rules is not consistent with Martin's teaching of the use of the rule to test integrity of an object/application. Appellant's statement is unfounded. The Examiner did in fact make specific references to the cited portions of the reference the Appellant has already acknowledged the cited portions above.

Appellant's Argument

"The Office Action alleges that Martin teaches a method of performing general integrity checks using an application running on a data processing system at pages 133 and 138-142. While it is true that Martin does teach the use of integrity rules and states that these rules indicate that something must be true (page 137), Martin does not teach the specific use of integrity rules set forth in claims 1, 19 and 25 (hereafter only referred to as claim 1), as discussed below."

Examiner's Response

Appellant calls Martin's teaching of integrity checking "general integrity check". There is no such term of "general integrity check" in the industry. The Appellant has conveniently misrepresented the actual meaning of the integrity check of Martin. Martin is checking the integrity of an object/application.

Appellant's Argument

"The Office Action alleges that Martin teaches identifying a point in a unit of work where application state integrity is to be verified merely because Martin allegedly teaches object state rules at page 143. First, Martin does not teach or even suggest units of work as the feature is recited in the pending claims. A unit of work is a piece of business work which defines each business context in which it is carried out. While Martin teaches that business policies may be represented as rules (see page 133), Martin makes not mention or even suggestion regarding units of work. Furthermore, business policies are not the same as units of work. A business policy merely outlines the decision processes of a business system, it does not represent a unit of work that is to be performed."

Examiner's Response

Unit of work in the Specification on page 11 states one of ordinary skill in the art would understand this term (also noted in the FAOM). The rejection set forth the understanding by the art rejection. The Appellant left this term open to interpretation and the Examiner did just that interpret it. The Appellant has failed to provide a technical reason why the term should be given any different technical interpretation. Appellant's action is tantamount to sitting back and saying "No, that's not it", without having clearly stating in technical terms the difference during the active prosecution of this application. Since, the Appellant left the door open to interpretation the Appellant early in the prosecution history should have provided the technical reason why a difference exists. Appellant remains in a lazy posture of saying as little as possible but saying the rejection is not proper. Mere allegation of patentability is not a technical distinction. Unit of work is shown in the rejection as running a procedure/method or Rule. The majority of the Appellant's arguments hinge on the term "unit of work" which they have failed to distinguish in technical terms and support from the Specification.

Appellant's Argument

Art Unit: 2124

“Because Martin does not teach units of work, Martin cannot be found to teach or even suggest identifying a point in a unit of work where application state integrity is to be verified. While Martin teaches object state rules on page 143, all that is stated is “Object state rules are identified in object-structure analysis. They are associated with diagrams, such as the data structure diagram, object-relationship diagram, or composed-of diagram.” It is not seen how such a general statement somehow teaches the very specific feature of identifying a point in a unit of work where application state integrity is to be verified, as recited in claim 1. The description of object state rules on page 143 of Martin does not make any mention of units of work, let alone identifying a point in a unit of work where state integrity is to be verified.”

Examiner’s Response

The Examiner’s position on Unit of Work is clear and need not be repeated. Again, the Appellant’s argument hinges on their failure to clearly and concisely traverse the interpretation of “unit of work”. The record is clear as to the interpretation give to this term.

The following is provided despite the glaring flaw in the Appellant’s arguments because it sheds light on other flaws of the Appellant. The Appellant states “Martin teaches object state rules on page 143, all that is stated is “While Martin Object state rules are identified in object-structure analysis. They are associated with diagrams, such as the data structure diagram, object-relationship diagram, or composed-of diagram.” It is not seen how such a general statement somehow teaches the very specific feature of identifying a point in a unit of work where application state integrity is to be verified, as recited in claim 1”.

The Appellant states says they don’t see how the statement teaches verifying application state integrity. The root of the lack of understanding seems to be at the conceptual level of programming and the failure to take the reference as a whole.

The reference clearly and repeatedly states diagrams are executable. In other words the diagrams the Appellant is referring to generate executable code. Furthermore, ability to call a method in object oriented programming is fundamental. One of ordinary skill should understand that to run an integrity check one would call a method. The Appellant is apparently protesting because the executable diagram that is cited teaches integrity checking but does not say it is run ? The point which the integrity check is run is the point where the state integrity is to be verified. It appears the Martin reference was not taken as a whole.

Appellant’s Argument

“Furthermore, even if it were interpreted that Martin teaches units of work, Martin still does not teach identifying a point in a unit of work where application state integrity is to be verified. As noted above, the general statement that there are object state rules, that they are identified in object-structure analysis, and that they may be associated with diagrams, has nothing to do with identifying a point in a unit of work where application state integrity is to be verified. The Examiner is reading features into the Martin reference that are simply not there in order to arrive at Appellants’ claimed invention having first had benefit of Appellants’ disclosure. In other words, the Examiner is engaged in hindsight reconstruction in which the Examiner is conjuring teachings from the reference that simply are not there.”

Art Unit: 2124

Examiner's Response

Appellant's arguments on "unit of work" and point in the unit of work with the executable diagrams are covered above. The Martin reference teaches the specific rule for performing an integrity check of an object/application. The fact that Martin states the diagrams are executable seems to have passed by the Appellant. The diagram represent code (units of work) and teach in the diagram (code) the calling of a specific rule of performing an integrity check on an object/application. It is apparent the Martin teaching is not understood as a result not having taken the reference as a whole.

Appellant's Argument

"The Final Office Action alleges that Martin teaches that a unit of work includes a plurality of participants merely because Martin allegedly mentions "three more rule windows that are linked to appropriate items in 00 diagrams" (page 143, last paragraph). It is not understood how linking rule windows to object oriented diagrams teaches a unit of work having a plurality of participants. There is no correlation between the feature recited in claim 1 and the cited portion of Martin. Linking rule windows to object oriented diagrams has nothing to do with a unit of work having a plurality of participants."

Examiner's Response

Again this is related to the Appellant's lack of understanding of the Martin reference. The key to Appellant's case is the critical term "unit of work " which has already been addressed above. Furthermore, *participants* was interpreted as methods in object technology. The Martin reference teaches the diagrams are executable. The ability to define RULE in a window is one way rules can be entered. One of ordinary skill should understand that objects have methods (some refer to them as operations). PARTICIPANTS from the FAOM were interpreted as methods. The Martin reference is clear on how the diagrams produce executable code and utilize editors such as the Rules Editor. The Martin reference is a teaching of OO-CASE tools which link the analysis, design and code generation in development environment with diagrams and editors. When taken as a whole the cited portions of the rejection teach the claim limitations.

Appellant's Argument

"The Final Office Action further alleges that Martin teaches obtaining rules for each of the participants in the unit of work merely because Martin provides example of rules associated with diagrams of object oriented analysis in Box 10.3 on page 144. Box 10.3 of Martin does not teach a unit of work, let alone obtaining rules for each participant in the unit of work. The only thing that Box 10.3 teaches is examples of rules for different types of object oriented diagrams. There is no teaching or even suggestion in Box 10.3 of using units of work, and definitely no teaching or suggestion of obtaining rules for each participant in a unit of work. Again, the Examiner is engaging in hindsight reconstruction by reading in teachings to the Martin reference that simply are not there."

Examiner's Response

This argument is redundant in view of the arguments above. The lack of understanding object oriented technology and how it is implemented in Martin to deploy OO-CASE tools where Rule editors and executable diagrams are utilized has left the Appellant arguing hindsight. This

Art Unit: 2124

argument is not persuasive. The arguments don't reflect the rejection and the interpretation of the terms. And not technical differences are provided. The same flaws over unit of work, diagrams and participants are present.

Appellant's Argument

"In addition, the Final Office Action alleges that Martin teaches running the rules obtained for each of the participants to verify the integrity of an application state merely because Martin allegedly teaches object state rules on page 143. The general teaching of object state rules does not provide any teaching or suggestion regarding running rules obtained for participants in a unit of work to verify the integrity of an application state. The general teaching of object state rules makes no teaching or suggestion of units of work, rules for participants in a unit of work, obtaining and running rules for participants in a unit of work, or that there is any correlation between rules for participants in a unit of work and an application state. There simply is no correspondence between the section of Martin cited by the Final Office Action and the features recited in claim 1."

Examiner's Response

The term "participant" in the first action and throughout the prosecution was given the meaning METHOD. This is part of an object by definition. Furthermore, the Appellant has never provided a technical reason why this interpretation is not correct just made general allegations of patentability. Since, OBJECT inherently have METHODS and performing a METHOD meets the definition of a UNIT OF WORK the rejection meets the claim limitations.

Appellant's Argument

"In summary, Martin has nothing to do with the invention recited in claim 1. While Martin provides a number of general teachings, they do not teach or suggest any of the features recited in claim 1. Furthermore, the Examiner appears to be stretching the interpretation of Martin in an attempt to reach Appellants' claimed invention beyond any reasonable interpretation of the reference. Such a stretch of interpretation is based on hindsight reconstruction using Appellants' own disclosure as a guide."

Examiner's Response

To believe Martin has nothing to do with the claimed invention one would need to believe the Appellant in the Specification in the section Description of Related Art was confused when they made explicit reference to the use of object oriented technology and proceed to refer to functionality that maps to both the cited portions of Martin and to other sections of Martin, which is why the Appellant received a 400 page reference.

Appellant's Argument

"Appellants do agree with the Final Office Action that Martin does not teach obtaining rules for each of the participants in a unit of work in response to determining that a unit of work is to be completed, as recited in claim 1. This is because Martin does not teach a unit of work, participants in a unit of work, or even that a unit of work may be completed. However, Appellants disagree with the Final Office Action regarding the allegation that Martin provides "endless possibilities" of conditions and operations such as write operations and that when to

Art Unit: 2124

perform a write operation and when not to perform a write operation somehow makes obvious this feature. The Final Office Action also cites Code Complete as teaching defensive programming which the Final Office Action somehow links with Martin to allegedly modify the Martin reference so that a determination that a unit of work is to be completed is somehow made obvious."

Examiner's Response

Appellant's "agreement" with the Final Office action appears to be that Martin does not explicitly teach control flow which is inherent in execution of programming languages. It is when control is returned after performing a unit of work.

Appellant's disagreement over whether Martin teaches endless possibilities of conditions or not is pointless since the reference teaches RULES and performing integrity testing of objects/applications. The Appellant is wavering over the teaching of terms. The record is clear to the meaning both in the Examiner's initial interpretation and the Appellant's own words the record is clear and the rejection's use of the terms. The Appellant has made general allegations but not technical distinction through out the prosecution history.

Appellant's Argument

"First, the Appellants cannot follow the Examiner's reasoning. It is not at all clear how a general teaching of conditions can be found to teach the specific feature set forth in claim 1 of a determination that a unit of work is to be completed. The Final Office Action states that the conditions of when to write and when not to write make this feature obvious. Appellants do not claim to be the first to invent determining if a unit of work is to be completed. Rather, Appellants claim obtaining rules for participants in the unit of work when it is determined that a unit of work is to be completed. Thus, the Final Office Action's example of a write condition has no bearing on the presently claimed features."

Examiner's Response

Examiner acknowledges the Appellant's inability to follow the reasoning of how Martin teaches OO-CASE tool and the specific feature of building RULES that specifically check integrity of objects/applications. The Martin reference when coupled with the Code Complete reference a common sense guide to programming techniques teach the claimed invention. Appellant's attack on an example given with testing the integrity of data before performing a "write" condition is piecemeal at best and appears to be an attempt to over narrow the rejection. The Examiner provided an example of a common sense test.

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Appellant's Argument

"Second, it is not clear how a general teaching of "defensive programming" may somehow be combined with the general teachings provided in Martin to arrive at the specific method set forth in claim 1. The Code Complete reference does not provide any teachings that

Art Unit: 2124

cure the deficiencies in Martin set forth above. In other words, the general teaching of "defensive programming" does not render obvious "identifying a point in a unit of work where application state integrity is to be verified, wherein the unit of work includes a plurality of participants; responsive to determining that the unit of work is to be completed, obtaining rules associated with each participant in the unit of work; and responsive to obtaining the rules, running the rules obtained for each of the participants to verify the integrity of an application state, according to the plurality of participants," as recited in claim 1."

Examiner's Response

In addition to the repeated arguments to terms. Appellant's states it is not clear how one would combine the OO-CASE tools of Martin with the programming technique of Code Complete. Yet Code Complete teaches common sense programming techniques when programming. In this rejection the programming construct of using an OO-CASE tool and the RULES feature is taught by Martin. Programmers and programming technique have provided means to provided integrity checking long before the filing of this application as taught by the references.

Appellant's Argument

"In addition, the IBM Computer Dictionary provides no additional teaching regarding any of the features of claim 1. That is, the combination of the IBM Computer Dictionary with Martin and Code Complete does teach or suggest any of the features in claim 1. None of these references have anything to do with the presently claimed invention recited in claim 1. The Examiner has taken extremely general teachings in a textbook and attempted to stretch these teachings beyond reasonable interpretation, and read additional teachings into the references that are not there, in an attempt to encompass the very specific method set forth in claim 1. Such a stretch of the teachings and reading in of additional teachings is not supported by the references and is based solely on hindsight reconstruction using Appellants' own disclosure as guide."

Examiner's Response

The statement of hindsight reconstruction is viewed more as an admission that the rejection meets the claim limitations. The Examiner showed common sense programming as taught by Code Complete a Microsoft Press publication on how to write sounds code when using the Martin teaching of Rules. The IBM Dictionary shows that many of the concepts implemented in the claim have been terms in the industry and are in the Assignee's own dictionary. The hindsight argument is unpersuasive.

Examiner's Position on Group I

The Appellant has made a late attempt to distance themselves from the meaning given throughout the prosecution to terms. And the only generalization is the Appellant's general allegation of patentability. It is clear that the Appellant is unable to make a clear technical distinction over the prior art of record. Attempts to argue show the Appellant not willing or unable to clearly state a different meaning of the terms and pointing to the Specification to prove the difference. Although, long winded the attack is nothing more than a general allegation of patentability. With patentability hinging on the arguments of claim 1 the Examiner has found claims 1, 7, 10, 12, 19, 24 and 25 unpatentable over the prior art of record.

Art Unit: 2124

Appellant's Argument for Group II - Patentability for claims 8, 22, 28 hinge on claim 8

With regard to claims 8, 22 and 28, neither Martin, Code Complete, or the IBM Computer Dictionary teaches or suggests detecting a commit for a unit of work. As set forth above, a commit in this context is a process for determining whether to commit the state changes made by the unit of work to the participants in the unit of work. The Final Office Action alleges that the mere teaching of having conditional rules in Martin somehow is the same as detecting a commit for a unit of work. Appellants respectfully disagree. Yet again, the Examiner is reading in teachings to the Martin reference that are not there. The general teachings of conditional rules does not teach or suggest detecting a commit of a unit of work.

As previously noted above, Martin does not teach a unit of work. Therefore, Martin cannot teach detecting a commit of a unit of work. Furthermore, Martin does not teach a commit of a unit of work. A general teaching of a condition rule does not provide any teaching or suggestion of a determination of whether to commit the state changes made by a unit of work to the participants in the unit of work, i.e. a commit of a unit of work.

In addition, claims 8, 22 and 28 include features that are similar to some of the features in claims 1, 19 and 25. Therefore, Martin, Code Complete and the IBM Computer Dictionary do not provide any teaching or suggestion of these features in claims 8, 22 and 28, as discussed above. With regard to claim 12, as previously noted, none of the references teach or suggest a unit of work, participants in a unit of work, or locating rules for the participants in response to activation of the unit of work to complete the unit of work.

In view of the above, Appellants respectfully submit that neither Martin, Code Complete, or the IBM Computer Dictionary, either alone or in combination, teach or suggest the features recited in claims 1, 8, 12, 19, 22, 25, 28. At least by virtue of their dependency on claims 1, 8, 12, 19, 22, 25 and 28, respectively, none of the references either alone or in combination teach or suggest the features set forth in dependent claims 2-7, 9-11, 13-18, 23-24, 26-27 and 29. Accordingly, Appellants respectfully request withdrawal of the rejection of claims 1-29 under 25 U.S.C. § 103(a).

Examiner's Response

The Applicant left the door open to interpret the term "units of work" in the Specification and the Examiner did just that with the rejection. The Appellant failed to provide a sound technical argument why the interpretation was wrong. The Specification on page 11 lines 24 – 25 fail to define the term and leave it open to interpretation by stating one of ordinary skill in the art would understand the term. The Examiner by rejecting the claim provided an interpretation. The Appellant has failed to provide anything more than mere allegation that the interpretation is different than some undefined meaning in the Specification. Appellant's repeated failures to make a sound distinction are not the fault of the Examiner.

The Examiner did not teach the detecting a commit of a unit of work with Martin. The argument is piecemeal and does not reflect the actual rejection. Please, review the actual rejection again.

Art Unit: 2124

The Appellant continues to make the allegation that the references do not teach; unit of work, participants in a unit of work, or locating rules for the participants in response to activation of the unit of work to complete the unit of work. Yet the rejection did clearly identify these terms and their interpretation. The Appellant has not provided any technical argument proving the interpretation is wrong and supporting such as position by pointing to the Specification for support.

The terms and the interpretations given are part of the written record. The Examiner listed them in the FAOM and accepted the Appellant's input in their own words. Attempts to distance themselves at the time Appeal from the record without technical argument is not persuasive.

Examiner's Position on Group III

Appellant's Argument for Group III - Patentability for claims 2, 3, 9, 20, 21, 23, 26, 27 and 29 hinge on claim 2

In addition, none of the references teach or suggest any of the specific features set forth in the dependent claims 2-7, 9-11, 13-18, 23-24, 26-27 and 29. The Final Office Action provides specific rejections of each of the dependent claims based on the combination of references discussed above. Each of these rejections is based on the same flawed interpretation and reading in of teachings discussed above and therefore, are traversed for similar reasons as set forth above.

In addition, because the dependent claims build off of the features recited in their respective independent claims, the deficiencies of the cited references are likewise applicable to the additional features set forth in the dependent claims. Moreover, none of the references teach or suggest any of the specific features recited in the dependent claims.

For example, because none of the references teach or suggest a unit of work or running rules for participants in a unit of work, as recited in claim 1, none of the references can be found to teach "responsive to a negative result obtained by running the rules, aborting the unit of work," as recited in claim 2. Moreover, none of the references teach or suggest committing a unit of work when a positive result is obtained by running the rules for the participants in the unit of work, as recited in claim 3. The Final Office Action alleges that these features are taught by the IBM Computer Dictionary as an abnormal termination and a commit. Appellants disagree that the aborting of a unit of work is the same as an abnormal termination. However, even if it were the same, the IBM Computer Dictionary does not teach aborting a unit of work or committing a unit of work based on a result obtained by running rules for a plurality of participants in a unit of work in response to a determination that a unit of work is to be completed, at a point in the unit of work where application state integrity is to be checked. This same distinction applies to claims 9, 20, 21, 23, 26, 27 and 29 which recite similar features in their respective claim terminology.

Examiner's Response

Arguments directed toward the independent claims were allegations of patentability and not persuasive. The rejection showed the use of return codes to indicate success or failure of an operation. The IBM dictionary showed the actions of commit or abort are common actions. The

Art Unit: 2124

rejection and the example the Examiner provided showed the common sense of not performing a write operation if a operation prior had failed.

The same flawed arguments to unit of work, plurality of participants and point in the unit of work are present in the argument for the dependents claims.

Examiner's Position on Group III

The Appellant has tied patentability to the same terms of unit of work, plurality of participants and point in the unit of work as in the independent claims. The Appellant failed to provide for the record a sound technical argument with their attempt to distance themselves from the interpretations.

Furthermore, the rejection teaches a combination of common sense programming techniques combined with an OO-CASE tools ability to perform integrity checks of objects. Appellant's arguments are not persuasive.

Appellant's Argument for Group IV

With regard to claim 4, the Final Office Action alleges that the features of each participant having an associated name and obtaining rules based on the name associated with the participant is taught by Martin in that Martin teaches linking rules to diagrams. This in no way has anything to do with obtaining rules for a plurality of participants in a unit of work based on the names associated with the participants. This same distinction applies to claim 16 which recites a similar feature in its respective claim terminology.

Examiner's Response

Since PARTICIPANT has been interpreted as a method since FAOM and methods inherently have names it is not possible to distinguish the claimed invention based on this argument. Furthermore, the repeated flaw of the Appellant of missing the fact that the diagram generate executable code links them to the code. Rules are code and can not be separated from the rejection.

Examiner's Position on Group IV

Appellant's arguments are wholly unpersuasive.

Appellant's Argument for Group V

Regarding claim 5, the Final Office Action alleges that the features that each participant is an object and wherein the name associated with the object is the class name of a participating object is taught by Martin in box 10.3 as the "customer." Appellants are at a loss as to what the Examiner's reasoning is. While "customer" may be a class name, there is nothing in Martin, and especially box 10.3, that teaches participants in a unit of work being objects whose names are class names and wherein the names of the objects that are participating in the unit of work are used to obtain rules that are run to verify application state integrity. This same distinction applies to claim 17 which recites a similar feature in its respective claim terminology.

Examiner's Response

Again the term *participant* since FAOM has meant method. The Appellant should review the meaning of a constructor in a programming language such as C++ . The constructor is

Art Unit: 2124

inherent and the name of the class has a constructor with the same name to instantiate an object of that type. Appellant's arguments are neither reflective of terms in the art or terms of the prosecution.

Examiner's Position on Group V

Appellant's arguments are wholly unpersuasive.

Appellant's Argument for Group VI

With regard to claim 6, the Final Office Action alleges that the features of a unit of work being associated with a type and each participant being associated with a name, and the step of obtaining rules is based on the name of the participant and the type of the unit of work is taught by Martin at pages 136, 138-140 and 144 because Martin allegedly teaches rules linked to diagrams. There is nothing in these sections, or any other sections, of Martin that teaches types of units of work or names of participants in a unit of work. Moreover, there is nothing in Martin that remotely even resembles obtaining rules for participants in a unit of work based on the name of the participant and the type of unit of work. This same distinction applies to claim 18 which recites a similar feature in its respective claim terminology.

Examiner's Response

The argument directed to unit of work of a type is met by the rejection. A method is part of an object. Objects inherently have a type. The Participant as interpreted inherently have names to be executed. This is foundation knowledge to object oriented technology. When one steps back from these pieces and looks at the definition of an object it should be clear to one of ordinary skill in the art that the methods are associated with objects. These limitations can not be separated from the enabling technology of object oriented technology.

Examiner's Position on Group VI

Appellant's arguments are wholly unpersuasive.

Appellant's Argument for Group VII

Regarding claim 11, the Final Office Action alleges that the feature of each rule associated with a unit of work having available for use each participant within the unit of work is taught by Martin simply because Martin teaches structures of rules. However, a general teaching of a rule structure, as set forth in Martin does not make any teaching or suggestion as to a rule being associated with a unit of work and that the rule associated with the unit of work may be used with each participant within the unit of work.

Examiner's Response

Again the Appellant has brought up the unit of work and the association of a unit of work (methods/rule) and the ability to call another method.

Examiner's Position on Group VII

Appellant's arguments are wholly unpersuasive.

Appellant's Argument for Group VIII

Art Unit: 2124

Regarding claim 13, the Final Office Action alleges that the feature of activation of a unit of work control point being initiated by a commit instruction to the unit of work is taught by Martin on pages 140-142. There is nothing on pages 140-142 that even mentions a unit of work, let alone a unit of work control point being activated by a commit instruction.

Examiner's Response and Position on Group VIII

The unit of work argument has already been answered. The commit was taught and shown in the Examiner's example. Commit operations are a common operation and part of the IBM dictionary. The ability to test the integrity prior to the commit operation completes the grounds of rejection and is absent in the argument above. This argument is viewed as piecemeal and incomplete.

Appellant's Argument for Group IX

With regard to claim 14, the Final Office Action alleges that the feature of a control point identifying applicable rules for all of the participants in the unit of work is taught by Martin at pages 140-141 because Martin teaches the ability to have conditionals. Conditional rules have nothing to do with the feature of a control point identifying rules that are applicable to all of the participants in a unit of work. There is no correlation between the Examiner's "conditionals" and the features recited in claim 14.

Examiner's Response

The control points are part of the diagrams that produce executable code. Page 140-141 teach anatomy of the control points. The operations are methods (participants).

Examiner's Position on Group IX

Appellant's arguments are wholly unpersuasive.

Appellant's Argument for Group X

Regarding claim 15, the Final Office Action alleges that the feature of a control point applying applicable rules to a portion of the participants in a unit of work is taught by Martin at pages 140-141 because Martin teaches the ability to have conditionals. Conditional rules have nothing to do with the feature of a control point identifying rules that are applicable to a portion of the participants in a unit of work. There is no correlation between the Examiner's "conditionals" and the features recited in claim 15.

Examiner's Response

The argument over applying control points is similar to the argument above where the Appellant seems to miss the diagram generate executable code and are linked to the terms that in question.

Examiner's Position on Group X

Appellant's arguments are wholly unpersuasive.

Appellant's Additional Arguments

Appellant states the following:

Art Unit: 2124

“ Examiner's Response to Above Arguments and Appellant's Rebuttal

In response to the above arguments, the Examiner, in the Final Office Action, alleges:

- 1) Martin provides teachings on Object Oriented CASE tools - Rules and common programming constructs (Final Office Action, page 39);
- 2) Appellants allegedly admit that the Martin reference teaches "units of work" (Final Office Action, page 40);
- 3) Martin teaches a Rule for integrity checking and rules may be attached to any diagram (as in any point) and mentions the diagrams are executable (Final Office Action, page 41);
- 4) Any judgment of obviousness is necessarily a reconstruction based on hindsight reasoning (Final Office Action, pages 41-42);
- 5) The Martin reference, when taken as a whole, teaches object oriented CASE tools for enterprise solutions and enterprises have many participants (categories of objects in Chapter 6 and interaction among objects in Chapter 7) (Final Office Action, page 42);
- 6) When taken as a whole, Martin teaches "actors" being defined and rules associated with the tasks of the participants on page 144 (Final Office Action, page 43);
- 7) participants are objects to which the integrity rules of Martin may be applied (Final Office Action, page 44);
- 8) In re Graves supports grounds of obvious rejections if an artisan of ordinary skill would have the knowledge at the time of invention on how to implement the basics of the tools of the trade as taught by Martin (Final Office Action, page 47);
- 9) The anatomy of a rule is demonstrated with common programming constructs (Final Office Action, page 48);
- 10) The Martin reference teaches the checking of integrity and Code Complete reference teaches common sense is to be used. The IBM dictionary underscores the industry use of the terms which took years to culminate into a denotation and published in a computer dictionary (Final Office Action, pages 49-50); and
- 11) One of ordinary skill in the art should know that the calling of a function transfers control and the completion returns control. Furthermore, the ability to check a return code in programming is grossly old and well known (Final Office Action, page 51).

The above responses by the Examiner may be summarized in one basic misplaced concept - the Examiner believes that any general teaching renders obvious all specific features the Examiner may argue are related, no matter how remotely related they may be. As a result, simply because Martin generally teaches rules and integrity checking rules, Martin now miraculously teaches all of the specific features set forth in Appellants' claims. This is clearly an incorrect approach to examination of patent claims. If such a notion were valid, all computer program claims would be obvious simply by virtue of the fact that there are textbooks that teach the basic constructs of programming for performing functions which could possibly be combined to achieve a claimed computer program, even though such combinations are not taught or suggested by the textbook. The Examiner seems to believe that simply because basic constructs could be combined to arrive at a claimed feature, that the mere possibility in itself is the suggestion to perform the combination. Thus, using the Examiner's reasoning, simply because Martin teaches object oriented programming, any program that operates in an object oriented environment is now obvious in view of Martin simply because the basic constructs in Martin could possibly be combined to generate any object oriented program, even though such

Art Unit: 2124

combinations are not taught or suggested by Martin. This is clearly not the case.”

Examiner's Response

First, the list of arguments is not specifying any specific claim. The Appellant's summary of the rejection and reference is not complete but this list of eleven points. The record speaks for the actual rejection and prosecution history. To refer to this list as a Summary would be to make a lazy and incomplete pass at the prosecution of this case.

The Appellant has been consistent in attempting to hide from interpretation of terms and providing a technical reason why the claimed invention is different. Visibly absent are technical arguments with merit that draw support for their position from the Specification. The basis of the arguments are refusal to define terms, attorney allegation of patentability and attack of the references (such as calling the Martin reference a general teaching). To highlight the Appellant's attack on Martin one only need to look at the last couple sentences above where the Appellant says *“Thus, using the Examiner's reasoning, simply because Martin teaches object oriented programming, any program that operates in an object oriented environment is now obvious in view of Martin simply because the basic constructs in Martin could possibly be combined to generate any object oriented program, even though such combinations are not taught or suggested by Martin. This is clearly not the case.”* The Examiner explicitly taught integrity checking of an object/application using the Martin reference. No where in any of the Examiner's actions does the Examiner state any Object Oriented Programming general constructs teach the invention. The Examiner cited where integrity checking is performed. The remaining arguments share this format of mere allegation and lack of technical arguments with support from the Specification and clear and concise definition of terms for the record.

Appellant's Argument

All of the Examiner's responses to Appellants arguments are rooted in this erroneous concept. The Examiner as much as explicitly states this when the Examiner refers to in re Graves (see response # 8 above) as allegedly supporting the proposition that the grounds of an obvious rejection are proper if an artisan or ordinary skill would have the knowledge at the time of invention on how to implement the basics of the tools of the trade as taught by Martin (Final Office Action, page 47). The Examiner is misapplying in re Graves. While one of ordinary skill in the art might arguably have the knowledge provided in Martin with regard to general concepts regarding object oriented programming, rules, and integrity checking rules, the mere knowledge is not sufficient without a suggestion or incentive to modify or combine this knowledge in such a way as to arrive at Appellants' claimed invention. What would prompt one of ordinary skill in the art to combine and modify such knowledge to arrive at the specific features in Appellants' claimed invention? There is nothing in Martin that explicitly teaches or suggests the features of the present invention, as discussed above.

Examiner's Response

It is important to note it only took the practice of testing a return code for the Appellant to state the Examiner used hindsight reconstruction. In re Graves is very clear about establishing what one of ordinary skill in the art would have known. The rejection of using a RULE to check the integrity of an application/object and test the result is within ability of one of ordinary skill in the art at the time of invention.

Art Unit: 2124

Appellant's Argument

To reduce the Examiner's position to its basic premise, the Examiner seems to believe that Appellants are merely claiming integrity checking rules in general and thus, the general teaching of integrity checking rules in Martin is used as a basis for rejecting all of claims 1-29. However, Appellants are not merely claiming any integrity checking rules but rather, as recited in claim 1, for example, a very specific mechanism for running rules for each of a plurality of participants in a unit of work in order to verify the integrity of an application state, in response to a determination that a unit of work is to be completed, at a point in the unit or work where application state integrity is to be verified. Martin does not teach such a mechanism. All that Martin teaches is that integrity checking rules may be used and attached to diagrams. There is nothing in Martin that teaches determining a point in a unit of work where the integrity of an application state is to be verified, determining if a unit of work is to be completed, obtaining rules for each of a plurality of participants in the unit of work in response to a determination that the unit of work is to be completed, and running the rules for each of the participants to verify the application state. Rather than finding a reference that explicitly teaches or suggests these specific features of the claimed invention, the Examiner insists on citing a general text book and making tenuous associations of teachings in Martin with claimed features when those teachings in Martin have nothing to do with the features of the pending claims.

Examiner's Response

The Examiner in the record provided interpretations consistently of the terms from the FAOM to the Examiner's Answer. The rejection was based on prior art not a "basic premise". The Examiner did not receive technical arguments and clear and concise arguments on terms with support pointing to the Specification. Instead, we have drama with phrases such as "tenuous associations of teachings" without technical arguments.

Appellant's Argument

With regard to the Examiner's response that Martin teaches object oriented CASE tools and common programming constructs (see response #1 above), Appellants are not in disagreement. Appellants' position is that even though Martin teaches such common programming constructs, these are not the same as the features recited in the claims. These common programming constructs do not teach or suggest identifying a point in a unit of work where application state integrity is to be verified, obtaining rules for a plurality of participants in the unit of work in response to a determination that the unit of work is to be completed, or running the rules for each of the participants to verify integrity of an application state. The mere general teaching of integrity checking rules does not rise to the level required to obviate these features.

Examiner's Response

The Examiner repeatedly provided the interpretation of identifying a point in a work unit and state integrity checking in the prior actions and above.

Appellant's Argument

Art Unit: 2124

Regarding the Examiner's response that Appellants allegedly admit that the Martin reference teaches "units of work" (see response #2 above), Appellants in no way admitted that Martin teaches units of work. Appellants' acknowledgment that Martin teaches that business policies may be implemented as rules is in no way an admission that Martin teaches units of work. Business policies are not units of work that are to be performed and have a plurality of participants that operate to complete the unit of work. Rather, a business policy merely states the general decision making of an enterprise solution. There is no correlation between business policies and units of work in the Martin reference. Thus, Appellants maintain that Martin does not teach units of work, despite the Examiner's allegation.

Examiner's Response

This point is moot. The Applicant left the door open to interpret the term "units of work" in the Specification and the Examiner did just that with the rejection. The failure of the Appellant to prove a sound technical argument why the interpretation was wrong gave the appearance we were in agreement. In viewing the record maybe we can agree that the Appellant still has not provided a clear and concise argument why the Examiner's interpretation is wrong. This might be because the Specification on page 11 lines 24 – 25 fail to define the term and leave it open to interpretation to one of ordinary skill in the art. The Examiner by rejecting the claim provided an interpretation. The Appellant has failed to provide anything more than mere allegation that the interpretation is different than some undefined meaning in the Specification. Appellant's repeated failures to make a sound distinction are not the fault of the Examiner.

Appellant's Argument

With regard to the Examiner's response that the general teaching of integrity rules and attaching integrity rules to diagrams renders obvious the feature of identifying a point in a unit of work where state integrity of an application is to be checked (see response # 3 above), Appellants respectfully disagree. Again the Examiner is taking general teachings and warping them to fit into the mold of Appellants' claims. There is nothing in a general teaching of attaching integrity rules to diagrams that can reasonably be interpreted as teaching identifying a point in a unit of work where application state integrity is to be checked. Where is the unit of work in a diagram?

Where are the plurality of participants in the unit of work illustrated in a diagram? Where is the obtaining of rules for the plurality of participants in the unit of work shown in the diagram?

Where is the running of rules for the plurality of participants in order to verify application state shown in the diagram?

Examiner's Response

In response to the Appellant's question of where are the plurality of participants. The Examiner interpreted the term participants as meaning METHODS in FAOM and this has been the meaning throughout the prosecution. Applicant seems to be imposing a one method per object limitation on the field of object oriented programming. This is not supported by the Specification. The argument indicated a lack of understanding of object oriented technology. Furthermore, Appellant's failure to take Martin as a whole seems to be why the Appellant does not understand that the diagrams in Martin produce executable code. With this apparent failure of the Appellant and the consistent use of terms by the Examiner the Appellant may want to

Art Unit: 2124

revisit the Martin reference. Also, noted that this argument is not directed toward any specific claim.

Appellant's Argument

Regarding the Examiner's response that any judgment of obviousness is necessarily a reconstruction based on hindsight reasoning (see response #4 above), Appellants agree. However, it is Appellants' position that the Examiner has not made a reconstruction based solely on the knowledge which was within the level of ordinary skill in the art at the time the claimed invention was made. The Examiner has based the rejection of claims 1-29 completely on knowledge gleaned only from Appellants' disclosure. Other than the general teaching of integrity checking rules, there is nothing in Martin that remotely resembles anything in the present claims. Thus, the Examiner, rather than finding a reference that teaches or suggests the features of the present claims, takes these general teachings of Martin and attempts to warp their interpretation to fit the mold defined by Appellants' claims. Such a process is based completely on hindsight. Therefore, while a judgment of obviousness must have some measure of hindsight, the finding of obviousness in the present case is solely based on hindsight. In re McLaughlin does not support the basis of the present rejection.

Examiner's Response

Not only do the Examiner and Appellant disagree on In re McLaughlin but Appellant's argument read on support for In re Graves which the Examiner also cited. Appellant's use of case law is wholly unpersuasive.

Appellant's Argument

"With regard to the Examiner's response that the Martin reference, when taken as a whole, teaches object oriented CASE tools for enterprise solutions and enterprises have many participants (see response #5 above), Appellants respectfully disagree that such a teaching renders obvious the features of the claimed invention. While Martin may teach objects and interaction of objects generally, there is nothing in Martin that teaches a unit of work having a plurality of participants and the identification of a point in the unit of work where application state integrity is to be checked. Moreover, there is nothing in Martin that teaches the obtaining of rules associated with each participant in a unit of work when it is determined that the unit of work is to be completed. Further, there is nothing in Martin that teaches running these rules for each of the participants to verify application state in accordance with the plurality of participants. The mere teaching of objects and objects being able to interact with one another does not rise to the level to obviate these features of the present invention."

Examiner's Response

If Applicant had taken reference as a whole the "many participants" which given the interpretation since FAOM means many methods would not have slipped by Appellant. The fact that Methods are an inherent part of Object technology would have been understood Martin dedicated Chapter 15 to the Creation of Methods. Then when you viewed the cited portions of Martin in the rejection you would know what a method looks like and would be able to spot more than one making it many.

Art Unit: 2124

Appellant's Argument

"With regard to the Examiner's response that Martin, when taken as a whole, teaches "actors" being defined and rules associated with the tasks of the participants on page 144 (see response #6 above), Appellants respectfully disagree. Page 144 of Martin only teaches examples of business rules associated with diagrams for object-oriented analysis. There is nothing on page 144 or any other section of Martin that teaches or suggests a unit of work having a plurality of participants, identifying a point in the unit of work where application state is to be checked, obtaining rules for the participants in response to a determination that a unit of work is to be completed, or running the rules for the participants to verify application state."

Examiner's Response

The real question is "WHY ARE WE TALKING ABOUT ACTORS?" No claim has this limitation present. The limitations that are present have already been addressed.

Appellant's Argument

Regarding the Examiner's response that participants are objects to which the integrity rules of Martin may be applied (see response #7 above), while the participants in a unit of work according to the present invention may be objects (see for example, pending claim S), Appellants disagree that the participants as defined in the present claims are objects to which integrity rules of Martin may be applied. First, even if integrity rules of Martin may be "applied" to objects, this still does not teach or suggest identifying a point in a unit of work where application state integrity is to be checked. Moreover, there is nothing in Martin that teaches that each of the participants in a unit of work have their own rules that are run to verify application state. Likewise, there is nothing in Martin that teaches the need to obtain such rules when it is determined that a unit of work is to be completed.

Examiner's Response

Integrity checks are operations/methods that must be called. What was the interpretation of PARTICIPANT since FAOM ? Answer – a METHOD. The execution of an integrity check in an object oriented environment means you must call a method/PARTICIPANT. All other limitations have been addressed.

Appellant's Argument

Regarding the Examiner's application of in re Graves (see response #8 above), this misapplication of case law is addressed above. With regard to the Examiner's response that the anatomy of a rule is demonstrated with common programming constructs (see response #9 above), the Examiner's response does not address Appellants' argument. That is, simply teaching that a rule is comprised of common programming constructs has nothing to do with determining whether a unit of work is to be completed and obtaining rules for each of the participants in the unit of work in response to determining that the unit of work is to be completed.

Examiner's Response

The determining whether a unit of work is completed has been covered numerous times and is repeated below. The teachings are well within one of ordinary skill at the time of invention.

Appellant's Argument

With regard to the Examiner's response that the Martin reference teaches the checking of integrity and Code Complete reference teaches common sense is to be used; and the IBM dictionary underscores the industry use of the terms which took years to culminate into a denotation and published in a computer dictionary (see response #10 above), Appellants respectfully submit that "common sense" and well known terms do not cure the deficiencies in Martin detailed repeatedly above. The use of "common sense" does not provide any teaching or suggestion to modify Martin to include an identification of a point in a unit of work where application state integrity is to be checked, obtaining rules for a plurality of participants in a unit of work in response to a determination that the unit of work is to be completed, or running the rules for each of the participants to verify the application state. Well-known terms, likewise, do not provide any of these teachings.

Examiner's Response

This attack is piecemeal. The Code Complete and IBM Dictionary were not used "to include an identification of a point in a unit of work where application state integrity is to be checked, obtaining rules for a plurality of participants in a unit of work in response" and "to verify the application state" (objects are applications), are taught by Martin. The limitation of testing the completion of a unit of work is an old technique that is explicitly taught by Code Complete. It is considered common sense to test if an operation was successful before you decided to perform an operation such as a write whose validity is dependent of the success of an operation. This was the example provided in the Final Action. Appellant called it limiting but failed to prove it is not "common sense" and supported by the cited portion of Code Complete.

Appellant's Argument

Regarding the Examiner's response that one of ordinary skill in the art should know that the calling of a function transfers control and the completion returns control; and furthermore, the ability to check a return code in programming is grossly old and well known (see response #11 above), Appellants respectfully submit that this has nothing to do with determining whether a unit of work is to be completed. Nowhere in Martin is it taught to determine whether a unit of work is to be completed. The allegation that checking return codes is the same as this claimed feature is simply wrong. Checking return codes has nothing to do with determining if a unit of work is to be completed and, in response to determining that the unit of work is to be completed, obtaining rules associated with each of the participants in the unit of work.

Examiner's Response

The Examiner made of record a very well known way to test for completion of an operation. In fact, the technical can tell if the operation was successful or a failure based on the return code. The Appellant says the invention is different but apparently can tell how.

Appellant's Summary

In summary, the Examiner has merely taken general teachings regarding basic concepts of object oriented programming as taught by Martin and has stretched and warped their interpretation in an attempt to fit the specific features of the present invention. There is nothing

Art Unit: 2124

in Martin, Code Complete, or the IBM Computer Dictionary that has anything to do with the present invention as recited in claims 1-29 other than a general teaching of integrity checking rules. There is nothing in any of these references that teaches or suggests a unit of work having a plurality of participants, identifying a point in the unit of work where application state integrity is to be checked, obtaining rules for each of the participants in the unit of work in response to a determination that the unit of work is to be completed, and running the rules for each of the participants in order to verify application state integrity.

Examiner's Response

As seen in the Appellant's Response (Paper # 11) the Appellant in their own words provided meaning to terms that the Examiner accepted and are a part of the written record. The Appellant provided meanings are consistent with the Teachings applied in the art rejection. One would need to stretch and warp technical teachings to attempt to distance themselves from the applied prior art of record. The Martin reference is more than a mere general teaching it provides discussion on companies products including screen shots and refers to work of the Assignee (IBM). The Code Complete reference teaches good programming technique. The IBM dictionary serves as a basic reference for basic terms and concepts which the record shows were made of record early in the prosecution and no technical reason on why the terms in the dictionary, interpreted by the Examiner or excellent detail provided by the Appellant were not related to the teaching of integrity checking rules.

Examiner's Final Remarks and Position

The prosecution history is clear. The Examiner went to great effort from the beginning to make the record clear. The terms were clearly defined as interpreted the Appellant's own words clarified terms. The Appellant has taken a lazy approach to saying those references and terms do not teach our invention without providing technical merit in the form of clear and concise technical argument. The Examiner has tried to find a way to allow the case but the lack of participation in the form of sound technical arguments makes maintaining the rejection the only option seen fit by the Examiner. If a twenty year right to exclude others from integrity checking in an object oriented programming environment is to be granted it will be granted by the Board of Patent Appeals and Interference (BPAI). The claimed invention is deemed obvious under 35 U.S.C. 103(a) in view of the prior art of record used in the rejection.

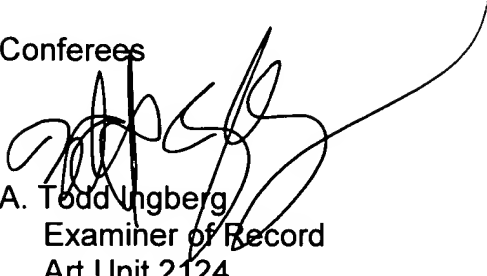
For the above reasons, it is believed that the rejections should be sustained.


Art Unit: 2124

Respectfully submitted,

Todd Ingberg
July 29, 2002

Conferees


A. Todd Ingberg
Examiner of Record
Art Unit 2124


B. GREGORY MORSE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

C. 
Patent Examiner

DUKE W YEE
CARSTENS YEE & CAHOON
P O BOX 802334
DALLAS, TX 75380